AFRL-VA-WP-TR-2005-3032

LARGE NUMBER OF AIR VEHICLES SIMULATION (LNAVSIM)
Phase II Extension

OR Concepts Applied
7032 Comstock Avenue, Suite 100
Whittier, CA 90602

MARCH 2005

Final Report for 31 May 2001 – 14 January 2005

THIS IS A SMALL BUSINESS INNOVATION RESEARCH (SBIR) PHASE II REPORT.

Approved for public release; distribution is unlimited.

STINFO FINAL REPORT

AIR VEHICLES DIRECTORATE
AIR FORCE MATERIEL COMMAND
AIR FORCE RESEARCH LABORATORY
WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7542

NOTICE

THIS TECHNICAL REPORT IS APPROVED FOR PUBLICATION.

/s/
EDWIN C. ALLEN, Project Engineer
Aerospace Vehicles Technology
Assessment & Simulation Branch

/s/
GARY K. HELLMANN, Chief
Aerospace Vehicles Technology
Assessment & Simulation Branch

/s/
BRIAN W. VAN VLIET, Chief
Control Sciences Division
Air Vehicles Directorate

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. REPORT DATE ***(DD-MM-YY)***	**2. REPORT TYPE**	**3. DATES COVERED** ***(From - To)***
March 2005	Final	05/31/2001 – 01/14/2005

4. TITLE AND SUBTITLE LARGE NUMBER OF AIR VEHICLES SIMULATION (LNAVSIM) Phase II Extension	**5a. CONTRACT NUMBER** F33615-01-C-3116 **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER** 605502
6. AUTHOR(S)	**5d. PROJECT NUMBER** A035 **5e. TASK NUMBER** **5f. WORK UNIT NUMBER** 0D
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) OR Concepts Applied 7032 Comstock Avenue, Suite 100 Whittier, CA 90602	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Vehicles Directorate Air Force Research Laboratory Air Force Materiel Command Wright-Patterson Air Force Base, OH 45433-7542	**10. SPONSORING/MONITORING AGENCY ACRONYM(S)** AFRL/VACD **11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)** AFRL-VA-WP-TR-2005-3032

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

This is a Small Business Innovation Research (SBIR) Phase II report. Report contains color.

14. ABSTRACT

Report developed under SBIR contract.

The Large Number of Air Vehicles Simulation (LNAVSIM) has provided ORCA the opportunity to research and design software tools to model, plan, and simulate large numbers of air vehicles. The primary goal of LNAVSIM has been to model air war campaigns. LNAVSIM provides an environment in which operators are able to sit at consoles capable of controlling multiple air vehicles (a pod). The pod operator is responsible for allocating the vehicles under his/her control and planning paths for the vehicles to accomplish assigned objectives.

15. SUBJECT TERMS

SBIR report, Coordinated Air Vehicle Movements, Mission Planning, Simulation, UAV, Flocking, Swarming, Birds.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT:**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** (Monitor)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Edwin C. Allen
Unclassified	Unclassified	Unclassified	SAR	20	**19b. TELEPHONE NUMBER** *(Include Area Code)* (937) 904-6521

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18

Table of Contents

Table of Figures

1 The LNAVSIM System

OR Concepts Applied (ORCA) began work on the LNAVSIM system in 2001 as an Air Force Research Laboratory (AFRL) sponsored Small Business Innovation Research (SBIR) project. Since that time all components of the LNAVSIM system continued to be enhanced.

1.1 Phase I

During Phase I, the first version of the architecture was developed to provide needed functionality. This architecture involved multiple components that run on multiple machines on different operating systems. These components were able to be added and removed at will to allow maximum flexibility in a study environment.

The Pod Operator Planning Interface (POPI) component is an Operator Vehicle Interface. This interface allows operators to perform target allocation, route planning, and analysis functions for multiple aircraft. The LNAVGEN Server performs these functions using OPUS 3 technology. A simulation component uses the OPUS 3 Simulation. There is also a Scenario Generator to construct the initial scenario. AnySim was developed to handle communication between different components using Simple Object Access Protocol (SOAP). A fairly new protocol at the time, it enabled sending function calls to other Web-enabled programs (Web services) using standard HTTP protocol and XML. The architecture uses this protocol because of its platform and language independence.

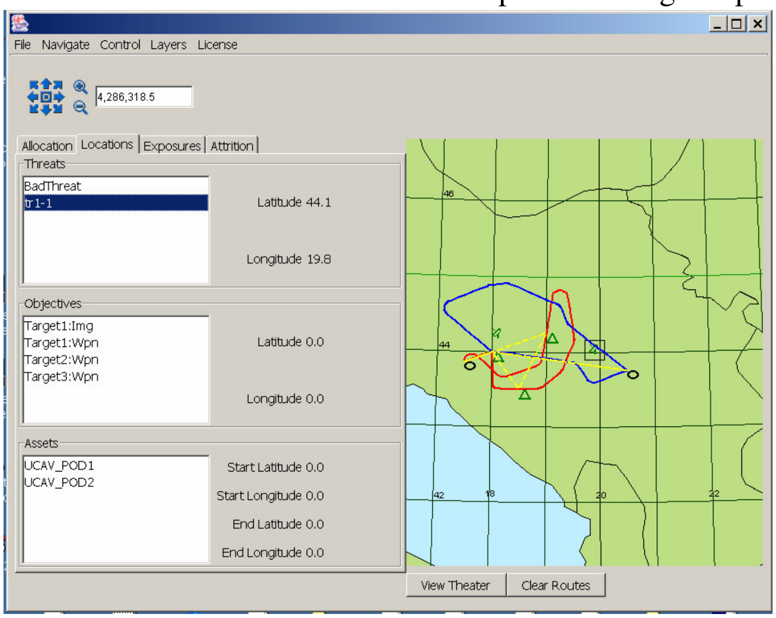

Figure 1. The POPI dialog

1.2 Phase II

During Phase II, the existing architecture was enhanced to provide greater flexibility. The LNAVGEN Server was modified to become a web service. This allowed the LNAVGEN Server to be used by the OVI, the Operator Vehicle Interface developed by AFRL for the LNAVSIM system. Additionally, the Broad Overseer of the Simulation and Scenario (BOSS) was developed as a graphical interface to the simulation. Because the Scenario Generator seemed limited as a separate component, the functionality of the Scenario Generator was incorporated into the BOSS. This resulted in greater flexibility for the LNAVSIM system. Using web services adds additional overhead to run the system because supporting middleware such as Apache Axis and Apache Tomcat server is required. This middleware was incorporated into the architecture, as well as the ability to run the LNAVSIM system without this middleware on a single machine. In that use case it uses RMI to handle intra-component communication.

1.3 Phase II extension

During the recently completed Phase II extension, ORCA continued enhancement of the architecture and components. The primary enhancement to the architecture was removal of the RMI. The advantages gained by using RMI were lost with the additional complexity needed to sustain this two-pronged strategy. Enhancements were made to the web services. Because of this, SOAP is now used exclusively for intra-component communication.

Another change was the addition of the Post Office. The Post Office performs the identical functions as the AnySim Interface in Phase I and Phase II. The AnySim Interface (AnySimIF) functionality was changed to allow any simulation to plug into the system. This made it much easier to change the simulation used in the LNAVGEN system.

The AnySimIF connects with a simulation-specific interface. This simulation-specific interface connects the native simulation to the AnySimIF and allows the simulation to be included without any changes to any other components. This allows greater flexibility since any simulation can be used without changes to any clients, the Scenario Generator, or the JFACC. Another significant change is the removal of the LNAVGEN Server. The Apache Axis middleware providing Web services in LNAVGEN was upgraded to support the Direct Internet Message Encapsulation (DIME) protocol. Fundamental differences in the C# and Java make sending complex objects through web service calls extremely difficult and, in some cases, impossible. The addition of DIME protocol made it possible to send these complex objects in their XML representations as attachments to SOAP calls. Therefore the LNAVGEN Server could be simplified to just a web service used the same way by all clients. This in turn led to retiring the LNAVGEN Server and replacing it with the OPUS 3 API SOAP Service. This service provides the same functionality as the LNAVGEN Server but with greater flexibility for the clients. The clients have access to controls to change the parameters for route generation and target allocation. This functionality allows clients greater capabilities. Additional data management functions were added to the OPUS 3 API SOAP service to allow easy testing and debugging of the interface. AFRL is currently developing the interface between OVI and the OPUS 3 API SOAP service.

1.3.1 OPUS 3 API SOAP service

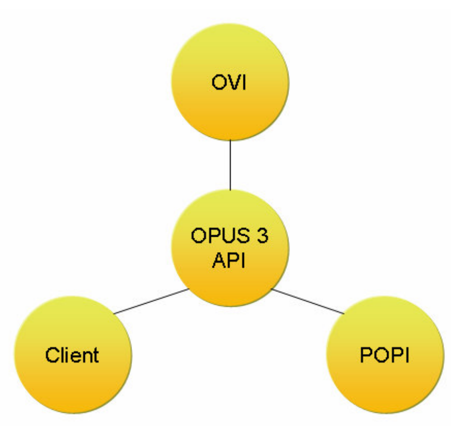

Figure 2. OPUS 3 API SOAP service

The OPUS 3 API SOAP service provides target allocation, route planning, and analysis. This service allows access to the OPUS 3 technology. It feeds OPUS 3 API SOAP service data in OPUS 3 XML format. The service then reads the data. Different functions (target allocation, route planning, and route analysis) can be called on the datasets, producing results which are then sent back to the client. Planning can be for either a full and complete route or a partial route. For partial routing the start point and end point are specified and only part of the existing route is changed. This is useful when the simulation is running and there is a popup threat necessitating dynamic replanning.

1.3.2 Post Office

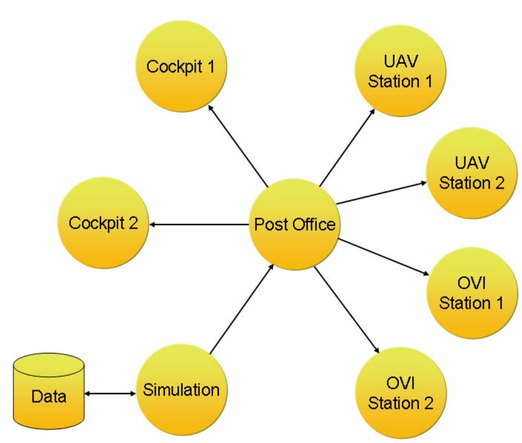

Figure 3. Post Office

The Post Office is also a web service. It is a repository for mail messages which contain communications between components. The Post Office places mail into a mailbox. When a component contacts the Post Office and requests mail, the Post Office delivers messages from all other components. This provides a central repository and greatly simplifies communication difficulties that would occur if all components communicated directly with the other components.

1.3.3 AnySim Interface (AnySimIF)

The AnySimIF is a reincarnation of the previous AnySimIF. The idea at first was to have the AnySimIF provide plug and play functionality for any simulation; however, after using the AnySimIF in previous versions of the LNAVSIM system, there was a need for an easier method to quickly and easily allow plug and play features of the simulations. The solution is the AnySimIF. It provides a standard set of functions and some functionality enabling the plug and play for simulation. The AnySimIF sends and receives mail. If a route is received it fires an event notifying the simulation. The AnySimIF applies the proper transformations to any data and sends the mail to the clients. For each simulation there must be a simulation-specific interface. This simulation-specific interface translates the data from simulation-specific objects into OPUS data objects. OPUS data objects are defined by the OPUS data DTD. These objects are used as the standard for the LNAVSIM system. Integration of any simulation into the LNAVSIM system requires a simulation-specific interface to handle the translation of the data. The simulation-specific interface calls the AnySimIF functions to send vehicle updates, data updates, or any other data updates. It also notifies the simulation when the simulation-specific interface receives notice from the AnySimIF of route updates.

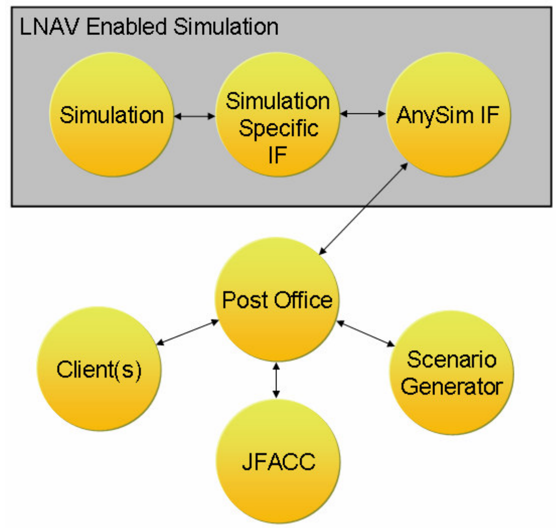

Figure 4. AnySim Interface

1.3.4 BOSS

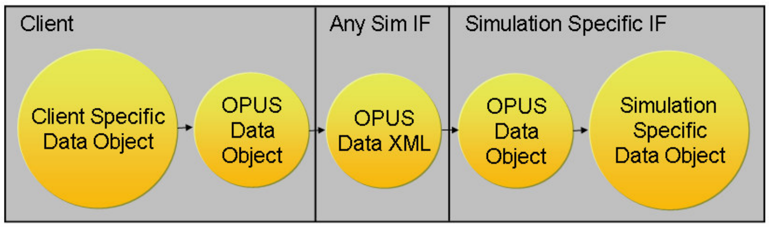

Figure 5. BOSS

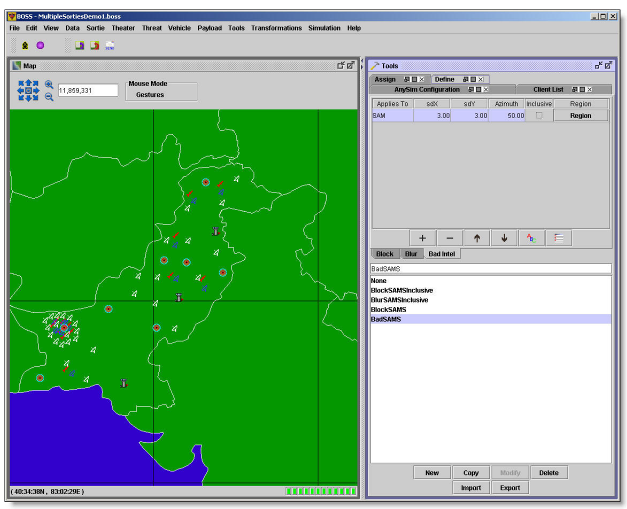

Figure 6. BOSS dialog

The BOSS contains the simulation-specific interface and the AnySimIF to integrate the OPUS 3 Simulation into the LNAVSIM system. The BOSS no longer contains scenario generation functionality, as the Scenario Generator was broken out into a separate component. The BOSS opens an OPUS 3 case, then configures the AnySimIF. This defines which clients are involved, which transformations are used for which clients, and if the JFACC will be used. Initial data is sent to the Scenario Generator to be transformed and sent to clients, who then generate routes to send to the simulation. Once started, the simulation sends vehicle status updates and any data changes that simulate popup threats. Route updates from the clients are incorporated into the simulation.

1.3.5 Scenario Generator

The Scenario Generator implements the initial scenario received from the AnySimIF, which includes the assets involved, threat status, and system data. The Scenario Generator may use transformations supplied by the AnySimIF or it can create its own transformations. It uses these transformations so that clients do not receive ground truth to use in routing. In the real world intelligence is not complete; therefore, transformations can be used to simulate this real world situation..

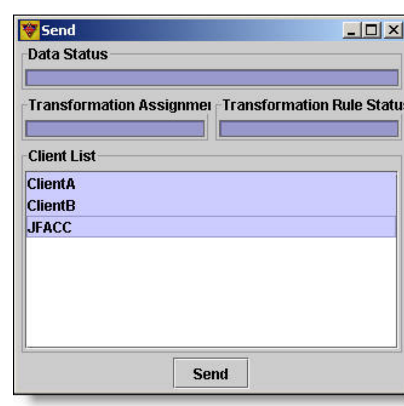

Figure 5. Scenario Generator dialog

1.3.6 JFACC

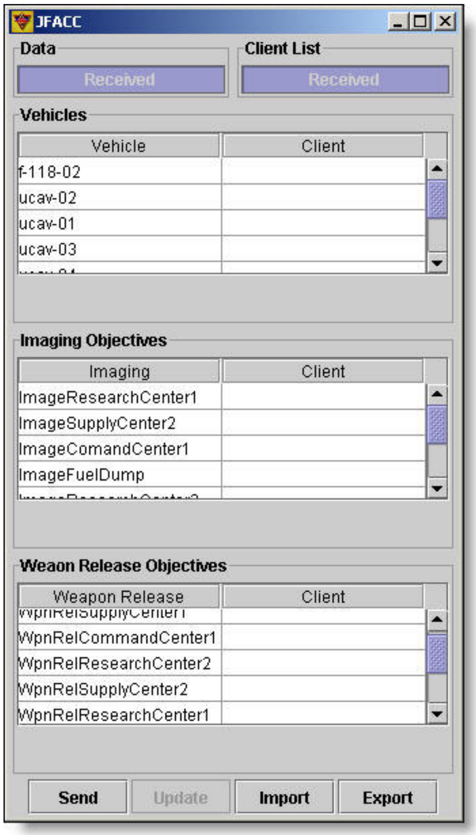

Figure 6. JFACC dialog

The JFACC component receives data from the Scenario Generator. JFACC can add additional objectives and targets to the data. JFACC then decides which assets and objectives to send to which clients. The JFACC can use OPUS allocation and router tools to help make these decisions. JFACC then sends the assets and objectives, along with their synergetic effect values, to the clients. JFACC is notified if additional assets are added to the case. JFACC may also be notified of popup threats and route changes. The JFACC can reallocate assets and create new objectives based on these data changes.

1.3.7 POPI

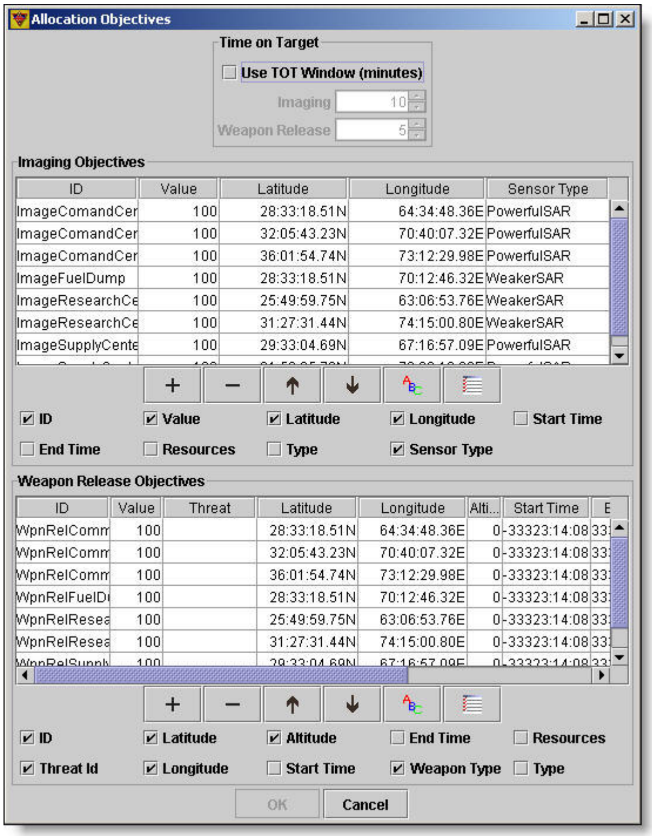

Figure 7. Allocation Objectives dialog

The POPI (Pod Operator and Planning Interface) is the component that allows an operator to control multiple vehicles. The operator can use the OPUS 3 API SOAP service to perform target allocation, route generation, and analysis. Once generated, the operator can send these routes to the simulation. The POPI also receives vehicle status and data updates from the simulation and reacts to these updates. If a threat changes, the operator can dynamically replan the route and send these changes to the simulation.

2 Potential future enhancements of LNAVSIM

2.1 OPUS 3 API SOAP service

Adding additional analysis services to the OPUS 3 API SOAP service would make more information available to help study pilot decision making. Currently, Figures of Merit (FOMs) are provided. Additional exposure reports and other analysis tools can be added to give the operator or pilot a better idea of what is happening in order to make better decisions.

The area of threat updates can be enhanced. When a threat update affects a vehicle, a report can be produced showing the effects of that threat change so the operator can decide whether to replan. Various levels of autonomy can be provided to specify if replanning should be automatic or involve user intervention. This can help additional studies of human factors with vehicle interfaces.

OPUS 3 target allocation is now the OPUS Synchronized Attack Planner. This planner offers improved target allocation performance. It provides the capability to create synchronized cooperative attacks for multiple aircraft and is recommended for use with between one and eight vehicles. It is not meant to be a force-level target allocator. It can allocate one objective among multiple vehicles. For example, if the objective is to drop twelve weapons on a SAM site and perform the imaging before and after, it could have one aircraft perform the initial imaging, another drop bombs, another drop four additional bombs, and a fourth vehicle perform the post damage assessment. The time of these attacks is constrained because some of the attacks need to occur before others, others at the same time, and others afterwards. The OPUS autorouter has also been enhanced to enforce time on targets. Previously if a time on target involved greater threat exposure it was avoided. Now the time on target is enforced even if it does cause greater threat exposure. These enhancements would help with study of heterogeneous aircraft groupings and cooperative missions.

Additionally the OPUS 3 API SOAP service could be changed into an interface providing target allocation, route generation, and analysis tools. Interfaces could be built for OPUS 3 PFPS, AFMSS, and JMPS. This would allow the same plug and play functionality for mission planning services as for simulations with the AnySimIF. Since the LNAVSIM system is highly flexible and there can be multiple clients, various clients can use various mission planning tools.

2.2 AnySimIF

The AnySimIF will probably experience the same evolution as the OPUS 3 API SOAP service as it progressed from the beginnings of the LNAVGEN Server to its current incarnation. As other simulations are incorporated into the system additional functionality will be discovered that makes integration of additional simulations into the AnySimIF and the LNAVSIM system much simpler and faster for developers.

Also, as the simulation-specific interface is developed for more simulations, various parts may be automated, requiring less work for developers. These may be in the form of additional enhancements to the AnySimIF or creating a generic simulation-specific interface that only needs a few functions to be built to integrate into a new simulation.

The AnySimIF lacks any simulation-driven functionality. At first glance, this would seem to make sense; however, if the simulation is running faster than real time and there is a popup threat this simulation should be slowed so operators can replan using real time, thereby simulating a real world situation. Once they finish replanning, the simulation should return to its previous speed if possible. This depends on the simulation: not all simulations can change their running speed. This functionality could be added to the AnySimIF and disabled in simulations that disallow it.

2.3 Make transformations that better mimic the real world

The transformations are currently a set of rules based on threat type and location. These rules define areas that block or otherwise distort a specific threat, a specific area, or a combination of both. This produces results that mimic intelligence seen in the real world. They have some shortcomings, however. As an aircraft flies it continues to gather intelligence using imaging sensors. Therefore transformations, instead of being based on a certain area, could be based on actual limitations of intelligence-gathering abilities. This could be simulated through satellite passovers or aircraft using imaging sensors. A Global Hawk can be sent to fly over a certain area to gather intelligence. The intelligence gathered by the aircraft's sensors can then be sent to clients simulating real world situations. The same can be done with satellite images: once the satellite images are gathered, there could be a time delay while image data is processed and sent to clients for use in additional routing. This more accurately reflects real world situations.

Another real world intelligence gathering problem is communications lines. Communication delays can be caused by a lag in the network system or part of the network being out of commission. Communication problems can be modeled in the transformations to make them more like the real world.

3 Conclusion

For the LNAVSIM project, ORCA developed tools that enable a single operator to control multiple aircraft (both manned and unmanned). Future development can build on previous work by:

- Enhancing components currently in the LNAVSIM system
- Incorporating additional relevant extant components
- Creating entirely new components as a result of further research and development

4 Acronyms

AFRL	Air Force Research Laboratory
API	Application Programming Interface
DIME	Direct Internet Message Encapsulation
DTD	Document Type Definition
HTTP	Hypertext Transfer Protocol
JFACC	Joint Force Air Component Commander
LNAVGEN	Large Number of Air Vehicles Generator
OPUS	ORCA Planning and Utility System
OVI	Operator/Vehicle Interface
RMI	Java Remote Method Invocation
XML	Extensible Markup Language